

RECEIVED
GROUP 230

1985 DEC 18 PM 2:01



C O P Y

IN THE UNITED STATES
PATENT AND TRADEMARK OFFICE

APPLICATION OF)

MARTIN G. REIFFIN)

FILED: SEPTEMBER 28, 1982)

GROUP 237

FOR: COMPUTER SYSTEM WITH)

REAL-TIME COMPILATION)

SERIAL NO. 425,612)

SUPPLEMENTAL AFFIDAVIT

County of Orange) ss.
State of California)

MARK WADSWORTH, being duly sworn, deposes and says that:

1. I am the affiant of the Wadsworth affidavit previously filed in the above application, and hereby supplement that affidavit as follows:

2. I have read and studied Paragraphs 21(f) through 21(k) inclusive on Pages 6-9 of the Office Action mailed September 19, 1985 in the above application, and the references designated in said paragraphs, and also the portion noted below of the Office Action of May 6, 1985.

3. Said Paragraphs 21(f) - 21(k) of the September 19, 1985 Action manifest four fundamental errors or misconceptions as follows:

(1) The failure to distinguish between the distinctly different operations of execution and compilation, so that the interrupted real-time execution of Benson et al. is confused with the interrupted real-time compilation of the subject invention;

(2) The failure to distinguish between real-time compilation concurrent with the entry of source code and the "line-by-line" incremental compiler scheme of the prior art discussed on Page 3 of the subject application and utilized in the Spangler reference.

(3) The failure to understand that the Benson et al. reference discloses merely "proposals" of goals and results to be achieved in a non-existent language proposed to be created in the future, and that the reference discloses neither hardware nor software for achieving the proposed goals and results;

(4) The misunderstanding of the concepts of concurrency and concurrent processes as used in computer science.

4. The following statements of the Office Action of September 19, 1985 appear to confuse the distinct operations of compilation and execution :

Page 6, Lines 15-18:
"The Benson et al. publication is...considered to show the general real-time compiling system advanced by the applicant."

Page 7, Lines 4-6:
"That Benson et. al. is directed to a compiler/computer system for real-time systems is an acknowledged fact..."

Page 7, Lines 9-11:
"...such [Benson et al.] system contemplated the equivalent of applicant's claimed 'contemporaneous real-time entry and compilation of source code'..."

5. That the term "real-time" is used in both the subject application and the Benson et al. proposals is possibly one source of the examiner's confusion. However, these two uses of the term have entirely different contexts and meanings. In the Benson et al. proposals the term "real-time" refers to the execution of the object code, whereas in the subject application the term refers to the compilation of the program source code.

6. That is, the Benson et al. proposals contemplate conventional compilation of programs which execute in real time, whereas the subject application is directed to real-time compilation of conventional programs which do not necessarily execute in real time.

7. In the subject application "real-time compilation" refers to compilation (translation) concurrent with the entry of the source code at the keyboard. In the Benson et al. proposals "real-time" refers to the execution (running) of the program concurrently with the operation of the mechanical or chemical process being controlled by the program.

8. The examiner's reliance upon "lines 9-27 of the right hand column on page 211 and 45-50 in the left hand column on page 212" of Benson et al. is misplaced. This portion of the reference merely says that it is "unlikely to be possible" to allow modification of the program while running, except for procedures declared in the outermost block. While this may be interpreted to mean that a limited form of compilation during execution may be possible, it does not even hint that compilation is contemplated during entry of the program at the keyboard.

9. The examiner's reliance upon "Lines, e.g. 37-44 in the right hand column on Page 204" as disclosing or suggesting a keyboard interrupt is untenable. The interrupting event contemplated here is one in the process being controlled in real time, not the entry of source code. Furthermore, the process being interrupted is the execution of the program, not its compilation. Here again, the Action has confused compilation with execution.

10. The following erroneous statement at Page 8, Lines 12-16 of the Office Action of September 19, 1985 manifests a failure to distinguish between the real-time compilation concurrent with the entry and editing of source code as disclosed in the subject application, and the "line-by-line" incremental compiler scheme of the prior art utilized in the Spangler et al. reference:

"The Spangler et. al. system was introduced into the record merely to positively show a console with a keyboard for key interruption of the operation of a compiler system for real-time entry of data/source code..."

11. None of the pages of the Spangler reference designated in the Office Action of September 19, 1985 discloses or suggests the interruption of the compiler by actuation of a keystroke, or by anything else. The Spangler abstract on Page 2, right column, Lines 11-12 states that the compiler operates "on a line-by-line basis." Since the language used in Spangler is BASIC which may be analysed both syntactically and semantically on a "line-by-line" basis, I understand the Spangler statement to mean that the reference employs a conventional incremental compiler as discussed at Page 3 of the specification of the subject

application. Such incremental compilers are not interrupted by a keystroke or otherwise, and such interruption would be pointless because the time required to analyse a single line of BASIC is almost instantaneous compared to the time required to strike a few keys of a keyboard.

12. Neither Benson et al. nor the pages and lines of Spangler et al. designated by the examiner contain a disclosure or suggestion of either:

- (1) the concurrent operation of compilation and source code entry; or
- (2) the interruption of compilation by the striking of a key at the keyboard.

13. The Benson et al. publication discloses merely "proposals" of goals and results to be achieved, and is devoid of any disclosure of either hardware or software for implementing the proposals so as to achieve these goals and results. To design and develop the required hardware and software to implement the proposals would take many man-years of effort which might well prove to be beyond the level of ordinary skill, and the resulting implementation might well prove to be impractical or inoperable.

14. The following erroneous statements on Pages 4,5 and 7,8 of the Office Action of May 6, 1985 manifest a misunderstanding of the meaning of the term "concurrent" as applied to concurrent processes in computer science:

"...no 'contemporaneous real-time entry and compilation of source code' appears possible with the passing of the control being claimed for the claimed 'compiling' and entering of the source code as set out in lines 4-12 do not occur at the same time (e.g. see item 39 of the March 11, 1985 affidavit by Wadsworth)... Claims 1, 6 and 11 and all claims dependent thereon would also appear to be misclaiming the 'contemporaneous entry

and compilation of source code' (e.g. performed 'at the same time')."

"The claimed 'concurrent' (e.g. 'at the same time') operation being claimed by the applicant is not shown by Benson et al. but then too it has been demonstrated that it is not supported by the applicant's specification and therefor it is considered that applicant has not established any right to depend on such claimed feature in arguing nonanticipation and/or nonobviousness of the cited prior art."

15. The above-quoted statements appear to founded upon the erroneous impression that the term "concurrent" as applied to processes means necessarily that the machine instructions of one process overlap those of another process so as to be executed "at the same time". This simultaneous execution of machine instructions may occur in a system with multiple processors operating in parallel, but the term "concurrent processes" is not limited to such systems and is also applied to the vast majority of systems having only a single processor. In the latter systems the machine instructions of one process are "interleaved in time" with those of another process, and two processes having this interleaving are also properly described as "concurrent processes". That is, "concurrency" exists "whenever the first operation of one process is started before the last operation of another process is completed."

16. As stated in the chapter entitled "Concurrent Processes" of the authoritative treatise "Operating System Principles" by Per Brinch Hansen of the California Institute of Technology, published by Prentice-Hall, Inc., at Page 57:

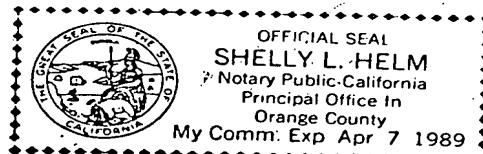
" Whether the individual operations of concurrent processes are overlapped or interleaved in time, or both (as shown in Fig. 3.3), is irrelevant. Whenever the first operation of one process is started before the last operation of another process is completed, the two processes are concurrent.

" In an installation where several processors work simultaneously, the machine instructions of concurrent processes can overlap in time. But if one processor is multiplexed among concurrent processes, the machine instructions of these processes can only be interleaved in time. The logical problems turn out to be the same in both cases; they are caused by our ignorance of the relative speeds of concurrent processes."

Mark Wadsworth

Mark Wadsworth

Subscribed and sworn to before me this 5th day of December, 1985.



Shelly L. Helm

Notary Public